RSA

CS/ECE 407

Today's objectives

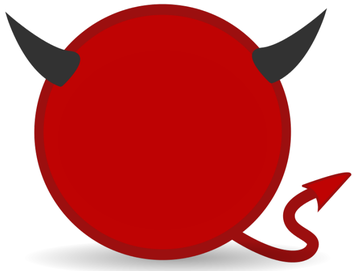
Describe another public-key crypto system: RSA

Understand the prime factoring problem

Construct RSA-based encryption and signatures

Alice

sk



Eve

Bob

pk

Public Key Encryption

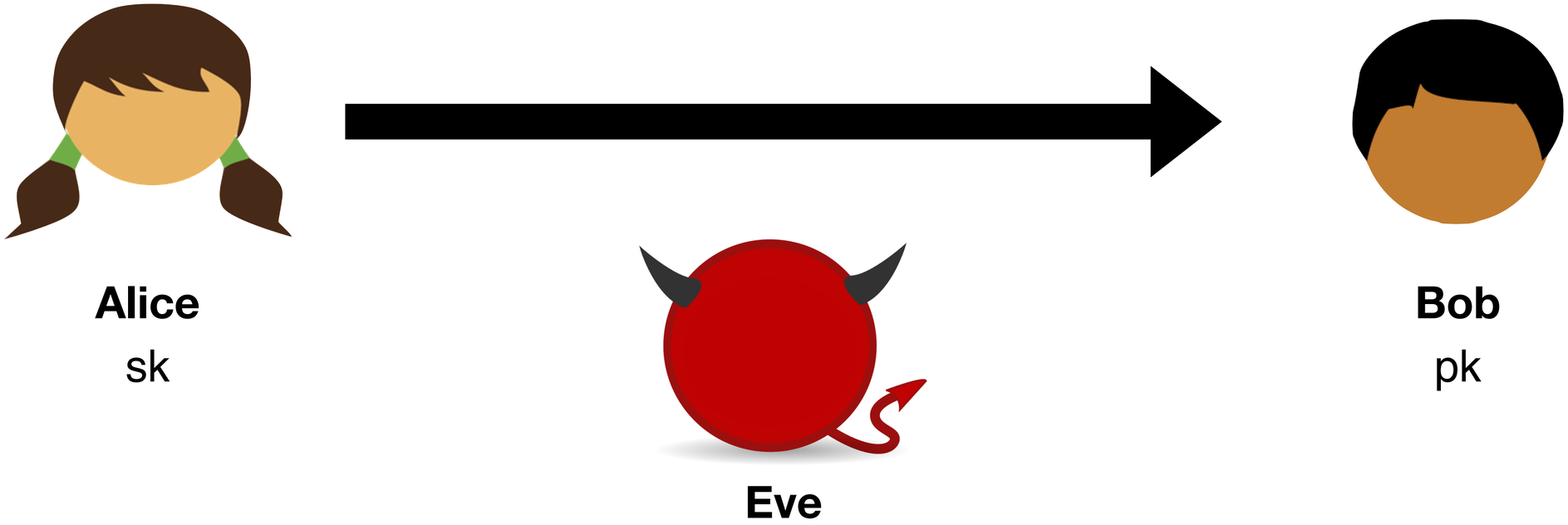
Anyone with pk can encrypt

Only one with sk can decrypt

Signatures

Only one with sk can sign

Anyone with pk can verify



Public Key Encryption

Anyone with pk can encrypt

Only one with sk can decrypt

ElGamal Encryption

Signatures

Only one with sk can sign

Anyone with pk can verify

Schnorr Signatures

Decisional Diffie-Hellman Assumption

Let Gen be an algorithm that defines a family of cyclic groups and their generators

We say the DDH problem is hard for that family if the following ensembles are indistinguishable:

$$\left\{ (\mathbb{G}, g, g^x, g^y, g^z) \;\middle|\; \begin{array}{l} (\mathbb{G}, g) \leftarrow \mathrm{Gen}(\lambda) \\ x \leftarrow \{0,\ldots |\mathbb{G}| - 1\} \\ y \leftarrow \{0,\ldots |\mathbb{G}| - 1\} \\ z \leftarrow \{0,\ldots |\mathbb{G}| - 1\} \end{array} \right\} \approx \left\{ (\mathbb{G}, g, g^x, g^y, g^{xy}) \;\middle|\; \begin{array}{l} (\mathbb{G}, g) \leftarrow \mathrm{Gen}(\lambda) \\ x \leftarrow \{0,\ldots |\mathbb{G}| - 1\} \\ y \leftarrow \{0,\ldots |\mathbb{G}| - 1\} \end{array} \right\}$$

RSA

Rivest, Shamir, Adleman

One of the oldest, most widely used public key systems

Based on the hardness of **factoring numbers**

RSA

Rivest, Shamir, Adleman

One of the oldest, most widely used public key systems

Based on the hardness of **factoring numbers**

Given product of two large primes $N = pq$ we believe it is hard to find p, q

Let p, q be primes

Let $N = p \cdot q$

$x^{-1} \bmod N$ exists if and only if $\gcd(x, N) = 1$

Number of elements in $\mathbb{Z}_N^*$

$(p-1)(q-1) = N - p - q + 1$

Let p, q be primes

Let $N = p \cdot q$

$x^{-1} \bmod N$ exists if and only if $\gcd(x, N) = 1$

Number of elements in $\mathbb{Z}_N^\times$

$\phi(N) = (p-1)(q-1) = N - p - q + 1$

$x^{\phi(N)} = 1 \bmod N$

RSA Assumption

$$\text{GenRSA}(1^\lambda) = \left\{ (N, e, d) \;\middle|\; \begin{array}{l} p, q \leftarrow \lambda\text{-bit primes} \\ N = p \cdot q \\ \text{choose } e > 1 \text{ s.t. } \gcd(e, \Phi(N)) = 1 \\ d = e^{-1} \bmod \Phi(N) \end{array} \right\}$$

We say the RSA problem is hard if for all polytime A the following is probability is negligible:

$$\Pr\left[x^e = y \bmod N \;\middle|\; \begin{array}{l} (N, e, d) \leftarrow \text{GenRSA}(1^\lambda) \\ y \leftarrow \mathbb{Z}_N^\times \\ x \leftarrow A(N, e, y) \end{array} \right] \leq \text{negl}(\lambda)$$

Key-Encapsulation Mechanism

An encryption scheme that works for random keys

```
KeyGen() → (pk, sk)

Encapsulate(pk) → (c, k)

Decapsulate(sk, c) → k
```

contrast with…

```
Enc(pk, m) → c
```

RSA KEM

```
KeyGen(1^λ):
  (N, e, d) ← GenRSA(1^λ)
  pk = (N, e)
  sk = (N, d)
  return (pk, sk)
```

$H : Z_N^\times \to \{0,1\}^\lambda$ is a random oracle

```
Encapsulate(pk = (N, e)):
  r ← Z_N^×
  c = r^e mod N
  k = H(r)
  return (k, c)
```

```
Decapsulate(sk = (N, d), c):
  r = c^d mod N
  k = H(r)
  return k
```

Existential Unforgeability under Chosen Message Attack (EUF-CMA)

```
(pk, sk) ← KeyGen()

key(): return pk

get(m):
  return sign(sk, m)

check(m, σ):
  return verify(pk, m, σ)
```

$\overset{c}{\approx}$

```
(pk, sk) ← KeyGen()
S ← empty-set

key(): return pk

get(m):
  σ ← sign(sk, m)
  S ← S ∪ {(m, σ)}
  return σ

check(m, σ):
  return (m, σ) ∈ S
```

RSA Signatures

```
KeyGen(1^λ):
  (N, e, d) ← GenRSA(1^λ)
  pk = (N, e)
  sk = (N, d)
  return (pk, sk)
```

$H : \{0,1\}^* \to Z_N^\times$
is a random oracle

```
Sign(sk = (N, d), m):
  σ = H(m)^d mod N
  return σ
```

```
Verify(pk = (N, e), σ, m):
  return σ^e = H(m) mod N
```

Today's objectives

Describe another public-key crypto system: RSA

Understand the prime factoring problem

Construct RSA-based encryption and signatures